Bu not defterinde, verilerin nasıl ölçeklendirileceğine ve normalleştirileceğine (ve ikisi arasındaki farkın ne olduğuna!) bakacağız.

Hadi başlayalım!

Get our environment set up[¶](https://www.kaggle.com/code/alexisbcook/scaling-and-normalization" \l "Get-our-environment-set-up" \t "_self)

Yapmamız gereken ilk şey kullanacağımız kütüphaneleri yüklemek olacak.

*# modules we'll use*

import pandas as pd

import numpy as np

*# for Box-Cox Transformation*

from scipy import stats

*# for min\_max scaling*

from mlxtend.preprocessing import minmax\_scaling

*# plotting modules*

import seaborn as sns

import matplotlib.pyplot as plt

*# set seed for reproducibility*

np.random.seed(0)

# Scaling vs. Normalization: What's the difference?[¶](https://www.kaggle.com/code/alexisbcook/scaling-and-normalization#Scaling-vs.-Normalization:-What's-the-difference?)

Ölçekleme (scaling) ve normalleştirme (normalization) arasında kafa karışıklığı yaşanmasının nedenlerinden biri, terimlerin bazen birbirinin yerine kullanılmasıdır ve daha da kafa karıştırıcı olması için, birbirlerine çok benzerler! Her iki durumda da, dönüştürülmüş veri noktalarının belirli faydalı özelliklere sahip olması için sayısal değişkenlerin değerlerini dönüştürüyorsunuz. Fark şudur:

* **Ölçeklemede**, verilerinizin **aralığını** değiştirirsiniz.
* **Normalleştirmede** ise, verilerinizin **dağılımının şeklini** değiştirirsiniz.

Şimdi bu seçeneklerin her biri hakkında biraz daha derinlemesine konuşalım.

# Scaling[¶](https://www.kaggle.com/code/alexisbcook/scaling-and-normalization" \l "Scaling" \t "_self)

Bu, verilerinizi 0-100 veya 0-1 gibi belirli bir ölçeğe sığacak şekilde dönüştürdüğünüz anlamına gelir. Veri noktalarının birbirinden ne kadar uzakta olduğunu ölçmeye dayalı yöntemler kullanırken (örneğin, **destek vektör makineleri (SVM)** veya **k-en yakın komşular (KNN)**) verileri ölçeklemek istersiniz. Bu algoritmalarla, herhangi bir sayısal özellikteki "1" birimlik bir değişikliğe aynı önem verilir.

Örneğin, bazı ürünlerin fiyatlarına hem Yen hem de ABD Doları cinsinden bakıyor olabilirsiniz. Bir ABD Doları yaklaşık 100 Yen değerindedir, ancak fiyatlarınızı ölçeklemezseniz, SVM veya KNN gibi yöntemler 1 Yen'lik bir fiyat farkını 1 ABD Doları'lık bir fark kadar önemli kabul edecektir! Bu, dünya hakkındaki sezgilerimize açıkça uymamaktadır. Para birimlerinde, para birimleri arasında dönüşüm yapabilirsiniz. Peki ya boy ve kilo gibi şeylere bakıyorsanız ne olacak? Kaç kilonun bir inç'e (veya kaç kilogramın bir metreye) eşit olması gerektiği tam olarak net değildir.

Değişkenlerinizi ölçekleyerek, farklı değişkenleri eşit bir zeminde karşılaştırmaya yardımcı olabilirsiniz. Ölçeklemenin neye benzediğini pekiştirmek için, uydurma bir örneğe bakalım. (Endişelenmeyin, sonraki alıştırmada gerçek verilerle çalışacağız!)

*# generate 1000 data points randomly drawn from an exponential distribution*

original\_data = np.random.exponential(size=1000)

*# mix-max scale the data between 0 and 1*

scaled\_data = minmax\_scaling(original\_data, columns=[0])

*# plot both together to compare*

fig, ax = plt.subplots(1, 2, figsize=(15, 3))

sns.histplot(original\_data, ax=ax[0], kde=True, legend=False)

ax[0].set\_title("Original Data")

sns.histplot(scaled\_data, ax=ax[1], kde=True, legend=False)

ax[1].set\_title("Scaled data")

plt.show()

Verinin şeklinin değişmediğini, ancak 0 ile 8 arasında değişen aralığın yerine artık 0 ile 1 arasında değiştiğini fark edin.

# Normalization[¶](https://www.kaggle.com/code/alexisbcook/scaling-and-normalization" \l "Normalization" \t "_self)

Ölçekleme (scaling) yalnızca verilerinizin aralığını değiştirir. Normalleştirme (normalization) ise daha köklü bir dönüşümdür. Normalleştirmenin amacı, gözlemlerinizi normal bir dağılım olarak tanımlanabilecek şekilde değiştirmektir.

**Normal dağılım:** "Çan eğrisi" olarak da bilinen bu dağılım, ortalamanın üstünde ve altında kabaca eşit sayıda gözlem bulunan, ortalama ve medyanın aynı olduğu ve ortalamaya daha yakın gözlemlerin daha fazla olduğu belirli bir istatistiksel dağılımdır. Normal dağılım, aynı zamanda Gauss dağılımı olarak da bilinir.

Google tarafından lisanslandı

Genel olarak, verilerinizin normal dağıldığını varsayan bir makine öğrenimi veya istatistik tekniği kullanacaksanız verilerinizi normalleştirirsiniz. Bunlara örnek olarak **doğrusal ayırma analizi (LDA)** ve **Gauss saf Bayes** verilebilir. (İpucu: Adında "Gauss" geçen herhangi bir yöntem muhtemelen normalliği varsayar.)

Burada normalleştirmek için kullandığımız yöntem **Box-Cox Dönüşümü** olarak adlandırılır. Şimdi, bazı verileri normalleştirmenin nasıl bir şey olduğuna kısaca bir göz atalım:

*# normalize the exponential data with boxcox*

normalized\_data = stats.boxcox(original\_data)

*# plot both together to compare*

fig, ax=plt.subplots(1, 2, figsize=(15, 3))

sns.histplot(original\_data, ax=ax[0], kde=True, legend=False)

ax[0].set\_title("Original Data")

sns.histplot(normalized\_data[0], ax=ax[1], kde=True, legend=False)

ax[1].set\_title("Normalized data")

plt.show()

Verilerimizin şeklinin değiştiğini fark ettiniz mi? Normalleştirmeden önce neredeyse L şeklindeydi. Ancak normalleştirmeden sonra daha çok bir çanın dış hatlarına benziyor (dolayısıyla "çan eğrisi").